

# BUNDESREPUBLIK DEUTSCHLAND



REC'D 06 OCT 2004  
WIPO PCT

## Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen:

103 49 581.9

Anmeldetag:

24. Oktober 2003

Anmelder/Inhaber:

ROBERT BOSCH GMBH, 70469 Stuttgart/DE

Bezeichnung:

Verfahren und Vorrichtung zur Umschaltung  
zwischen wenigstens zwei Betriebsmodi einer  
Prozessoreinheit

IPC:

G 06 F 9/48

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 23. September 2004  
Deutsches Patent- und Markenamt

Der Präsident  
Im Auftrag

Dzierzon

29.09.03 Sy/Ho

5

ROBERT BOSCH GMBH, 70442 Stuttgart

10

Verfahren und Vorrichtung zur Umschaltung zwischen wenigstens zwei Betriebsmodi einer Prozessoreinheit

Stand der Technik

15

Die Erfindung geht aus von einem Verfahren und einer Vorrichtung zur Umschaltung zwischen wenigstens zwei Betriebsmodi einer Prozessoreinheit sowie entsprechender Prozessoreinheit mit wenigstens zwei Ausführungseinheiten zur Abarbeitung von Programmen gemäß den Oberbegriffen der unabhängigen Ansprüche.

20

Solche Prozessoreinheiten mit wenigstens zwei integrierten Ausführungseinheiten sind auch als Dual-Core- oder Multi-Core-Architekturen bekannt. Solche Dual-Core- oder Multi-Core-Architekturen werden nach heutigem Stand der Technik hauptsächlich aus zwei Gründen vorgeschlagen:

25

Zum Einen kann damit eine Leistungssteigerung, also eine Performance-Steigerung erreicht werden, indem die beiden Ausführungseinheiten oder Cores als zwei Recheneinheiten auf einem Halbleiterbaustein betrachtet und behandelt werden. In dieser Konfiguration bearbeiten die zwei Ausführungseinheiten oder Cores unterschiedliche Programme respektive Tasks. Dadurch lässt sich eine Leistungssteigerung erzielen, weshalb diese Konfiguration als Leistungsmodus oder Performance-Mode bezeichnet wird.

30

Neben dem Einsatz als superskalare Prozessoren ist der zweite Grund eine Dual-Core- oder Multi-Core-Architektur zu realisieren, die Steigerung der Sicherheit, indem beide

Ausführungseinheiten redundant das gleiche Programm abarbeiten. Die Ergebnisse der beiden Ausführungseinheiten werden verglichen, und ein Fehler kann dann bei dem Vergleich auf Übereinstimmung erkannt werden. Im Folgenden wird diese Konfiguration als Sicherheitsmodus oder safety-mode bezeichnet.

5

Im Allgemeinen sind die beiden genannten Konfigurationen exklusiv auf der Dual- oder Multi-Core-Architektur enthalten, d. h. der Rechner mit den wenigstens zwei Ausführungseinheiten wird prinzipiell nur in einem Modus betrieben: Entweder dem Performance-Modus oder dem Safety-Modus.

10

Aufgabe der Erfindung ist es nun, einen kombinierten Betrieb einer solchen Dual- oder Multi-Core-Prozessoreinheit bezüglich wenigstens zwei Betriebsarten zu ermöglichen und dabei eine optimale Umschaltstrategie zwischen wenigstens zwei Betriebsmodi, also insbesondere zwischen Sicherheitsmodus und Leistungsmodus zu erzielen.

15

#### Vorteile der Erfindung

Es ist zum Einen aus Sicherheitsgründen eine redundante Ausführung der Programme respektive Tasks, also auch von Aufgabenprogrammen, Programmteilen, also 20 Codeblöcken oder auch einzelnen Befehlen erwünscht, aber andererseits aus Kostengründen das Bereithalten von vollständig redundanter Hardware bei der Ausführung der nicht sicherheitskritischen Funktionen nicht erstrebenswert. Dieser Zielkonflikt wird erfindungsgemäß durch eine optimierte Umschaltung zwischen wenigstens zwei Betriebsmodi bei einer Prozessoreinheit gelöst. So geht die Erfindung von einem Verfahren und einer Vorrichtung zur Umschaltung zwischen wenigstens zwei Betriebsmodi einer Prozessoreinheit mit wenigstens zwei Ausführungseinheiten sowie entsprechender Prozessoreinheit aus. Die Prozessoreinheiten können dabei zum Einen vollständige Cores, also vollständige CPUs sein oder aber auch in einem bevorzugten Ausführungsbeispiel wird nur das Rechenwerk dupliziert. Der Vorteil ist, wenn nur das 30 Rechenwerk (ALU) dupliziert wird und die anderen Komponenten der CPU durch andere Fehlerentdeckungsmechanismen abgesichert werden, benötigt die vorgestellte Schaltung zusätzlich weniger Chipfläche als eine vollständige Dual-Core-Architektur. Trotzdem kann aber mit dem erfindungsgemäßen Verfahren, also gleichermaßen für Doppel-CPU oder Doppel-ALU eine ausreichende Fehlerabdeckung im Sicherheitsmodus, und bei

nicht sicherheitsrelevanten Berechnungen eine deutliche Leistungssteigerung im Leistungsmodus erzielt werden. Die Erfindung geht also von einem Verfahren und einer Vorrichtung zur Umschaltung zwischen wenigstens zwei Betriebsmodi einer Prozessoreinheit mit wenigstens zwei Ausführungseinheiten zur Abarbeitung von 5 Programmen aus, wobei vorteilhafter Weise den Programmen wenigstens eine Kennung zugeordnet ist, welche eine Unterscheidung in die wenigstens zwei Betriebsmodi, also insbesondere Sicherheitsmodus und Leistungsmodus erlaubt und eine Umschaltung zwischen dem Betriebsmodi in Abhängigkeit von der Kennung erfolgt, so dass die Prozessoreinheit die Programme entsprechend des zugeordneten Betriebsmodus 10 abarbeitet.

Dabei umfasst der Begriff Programme auch Programmteile, also Codeblöcke, die sich über mehrere Programme vollständig oder ansteilig erstrecken über Aufgabenprogramme, die in den einzelnen Programmen enthalten sind oder von den Programmen gebildet 15 werden bis hin zu einzelnen Programmbefehlen, denen jeweils eine Kennung zugeordnet wird.

Dabei kann eine solche Kennungszuordnung zur Umschaltung zwischen den einzelnen 20 Betriebsmodi auf Funktionsebene, also insbesondere zur Steuerung von Betriebsabläufen bei einem Fahrzeug, Einsatz finden. Andererseits können auch vorteilhafter Weise Programme oder entsprechende Aufgabenprogramme, Programmteile oder Programmbefehle, die zu einem Betriebssystem der Prozessoreinheit gehören oder dieses 25 Betriebssystem darstellen, durch solche Kennungen dem entsprechenden Betriebsmodus zugeordnet werden.

Zweckmäßiger Weise werden bei der Abarbeitung der Programme die dabei entstehenden Zustände oder Ergebnisse auf Übereinstimmung verglichen, wobei bei Abweichung auf Fehler erkannt wird.

30 Dabei ist es besonders zweckmäßig, dass die Programme synchron abgearbeitet werden.

Vorteilhafter Weise ist die Kennung als wenigstens ein Bit ausgebildet, wobei eine solche Kennung zweckmäßiger Weise durch einen Programmbefehl, insbesondere durch einen

im Befehlssatz der Prozessoreinheit vorgesehenen Befehl, wie beispielsweise einem Schreibbefehl erzeugt wird.

5 Diese Kennung kann zum Einen dem entsprechenden Programm, Programmteil, Ausführungsprogramm oder Programmbebefhl zugeordnet sein oder aber in einen speziellen, vorgesehenen Speicherbereich eingeschrieben sein oder werden.

10 Somit kann abhängig von der Kennung eine optimierte Umschaltung zwischen zwei Betriebsmodi, insbesondere dem Leistungsmodus und dem Sicherheits-Modus bei einer Dual-Core-Architektur oder eine Architektur mit lediglich dupliziertem Rechenwert, also einer Doppel-ALU erfolgen.

15 Weitere Vorteile und vorteilhafte Ausgestaltungen ergeben sich aus der Beschreibung sowie den Merkmalen der Ansprüche.

#### Zeichnung

20 Die Erfindung wird im weiteren anhand der in der Zeichnung dargestellten Figuren näher erläutert.

25 Dabei zeigt Figur 1 und Figur 2 jeweils eine Prozessoreinheit mit dupliziertem Rechenwert, bei welcher die erfindungsgemäße Umschaltung durchführbar ist.

Figur 3 zeigt die Umschaltung aus dem Sicherheitsmodus in den Leistungsmodus und Figur 4 zeigt die Umschaltung vom Leistungsmodus in den Sicherheitsmodus.

30 Figur 5 zeigt anhand einer Anzahl von Codezeilen 500 die Zuordnung der Kennzeichen zu den Programmen, Programmteilen, Aufgabenprogrammen oder Befehlen.

#### Beschreibung der Ausführungsbeispiele

In den Figuren 1 und 2 der Zeichnung sind gleiche bzw. funktionsgleiche Elemente sofern nichts anderes angegeben ist - mit gleichen Bezugszeichen versehen worden. Die erfindungsgemäße programmgesteuerte Einheit sowie deren Komponenten wie

Mikrocontrollerkern (CPU), Speichereinheiten, Peripherieeinheiten usw. sind der besseren Übersichtlichkeit halber in den Figuren 1 und 2 nicht direkt dargestellt worden. Allerdings können die beiden Rechenwerke ALU A und Alu B ebenso kompletten Cores, also CPUs im Rahmen der Erfindung entsprechen, so dass die Erfindung auch für komplette Dual-Core-Architekturen Einsatz finden kann. Bevorzugt ist allerdings, dass nur das Rechenwerk dupliziert wird und die anderen Komponenten der CPU durch andere Fehlerentdeckungsmechanismen abgesichert werden.

In den Figuren 1 und 2 sind mit den Bezugszeichen 1 und 2 jeweils arithmetisch logische Einheiten (ALU) als Ausführungseinheiten bezeichnet. Eine jeweilige ALU-Einheit 1, 2 weist zwei Eingänge und einen Ausgang auf. In einem Testbetrieb können die zur Ausführung vorgesehenen Operanden direkt vom Bus 3 in die Eingänge der ALU-Einheiten 1, 2 eingekoppelt werden oder vorher in einem eigens dafür vorgesehenen Operandenregister 8, 9 abgelegt sein. Diese Operandenregistern 8, 9 sind direkt mit dem Datenbus 3 gekoppelt. Die beiden ALU-Einheiten 1, 2 werden also aus den gleichen Operandenregistern 8, 9 versorgt. Zusätzlich kann vorgesehen sein, dass die jeweiligen Operanden über den Bus bereits mit einer ECC-Codierung versehen sind, welche in den Registerbereichen 8a, 9a abgelegt werden. D. h. an allen Stellen in den Figuren 1 und 2, in denen ECC angegeben ist, können die Daten durch Einsatz eines ECC-Codes (Error Correction Code) abgesichert sein. Solche Methoden zur Fehlererkennung sind vielfältig, wobei die Grundvoraussetzung die Absicherung mit einem Fehlererkennungs- bzw. Fehlerkorrekturcode, also einer Signatur darstellt. Im einfachsten Fall kann diese Signatur nur aus einem Signaturbit, beispielsweise einem Parity-Bit bestehen. Andererseits kann die Absicherung auch durch komplexere ED-Codes (Error Detection) wie einen Berger-Code oder einen Bose-Lin-Code usw. oder auch durch einen komplexeren ECC-Code wie beispielsweise einen Hamming-Code usw. realisiert werden, um durch entsprechende Bitzahl eine sichere Fehlererkennung zu ermöglichen. Es kann aber auch als Codegenerator, beispielsweise eine Generatortabelle (fest verdrahtet oder in Software) verwendet werden, um bestimmten Eingangsmustern der Bits im Rahmen der Adresse ein gewünschtes Codemuster beliebiger Länge zuzuordnen. Damit kann, insbesondere durch die Korrekturfunktion, die Datensicherheit gewährleistet werden. Dennoch erfolgt im sicherheitskritischen Modus, also im Sicherheitsmodus SM eine redundante Abarbeitung der sicherheitskritischen Programme in beiden Ausführungseinheiten, also hier beiden

ALUs 1 und 2, wodurch Fehler in diesen durch Vergleich auf Übereinstimmung erfindungsgemäß aufgedeckt werden.

5 Die nicht sicherheitsrelevanten bzw. nichtsicherheitskritischen Programme oder Tasks respektive Programmteile oder Codeblöcke oder Befehle können zur Leistungserhöhung auf beiden Ausführungseinheiten verteilt berechnet werden, wodurch sich der Durchsatz und damit die Leistung erhöht. Dies erfolgt im sogenannten Leistungsmodus oder Performancemode LM.

10 Bei Einkopplung der jeweiligen Operanden in die ALU-Einheiten 1, 2 muss besonderer Wert auf die korrekte Dateneingabe gelegt werden. Werden z. B. die gleichen fehlerhaften Operanden in die beiden ALU-Einheiten 1, 2 eingekoppelt, ist ein Fehler am Ausgang der ALU-Einheiten 1, 2 nicht erkennbar. Es muss daher sichergestellt sein, dass zumindest eine der ALU-Einheiten 1 oder 2 einen korrekten Dateneingabewert erhält bzw. auch beide ALU-Einheiten 1, 2 unterschiedliche, jedoch falsche Dateneingabewerte erhalten. Dies wird dadurch sichergestellt, dass von zumindest einem Eingangswert einer ALU-Einheit 1, 2 eine Prüfsumme, also ein ECC-Code, wie oben erwähnt, gebildet wird. In einer eigens vorgesehenen Vergleichseinheit 5, 6 wird die ECC-Codierung 10a, 11a aus diesen zusätzlichen Datenregistern 10, 11 mit der ECC-Codierung 8a, 9a aus dem 15 ursprünglichen Quellregister 8, 9 verglichen. Optional können auch die Eingangsdaten aus den Registern 10, 11 mit denen aus den Quellregistern 8, 9 verglichen werden. Ergibt sich ein Unterschied in der ECC-Codierung bzw. bei den Operanden, dann wird dies als Fehler interpretiert und es wird ein Fehlersignal ausgegeben, unter Umständen angezeigt und unter Umständen korrigiert. Dieser Vergleich erfolgt vorteilhafter Weise während der 20 Bearbeitung der Operanden in den ALU-Einheiten 1, 2, so dass diese eingangsseitige Fehlererkennung und Fehlerkorrektur nahezu ohne Leistungsverlust einhergeht. Erkennt eine der Vergleichseinheiten 5, 6 einen Fehler, so kann die Berechnung innerhalb des nächsten Zyklus wiederholt werden. Dabei kann ein Schattenregister Einsatz finden, um 25 immer die Operanden der letzten Berechnung zu sichern, damit sie in einem Fehlerfall schnell wieder verfügbar sind. Auf die Bereitstellung eines solchen Schattenregisters kann allerdings verzichtet werden, wenn die jeweiligen Operandenregister 10, 11 erst wieder durch ein Freigabesignal aufgrund des Nichtvorhandenseins eines Fehlers erneut beschrieben werden. Im Falle eines Fehlers liefern die Vergleichseinheiten 5, 6 ein 30 Fehlersignal, wodurch die Operandenregister 10, 11 nicht erneut beschrieben werden.

5 Die ALU-Einheiten 1, 2 erzeugen ausgangsseitig jeweils ein Ergebnis. Die von den ALU-Einheiten 1, 2 bereitgestellten Ergebnisdaten bzw. deren ECC-Codierung werden in den Ergebnisregistern 12, 13, 12a, 13a abgelegt. Diese Ergebnisdaten und/oder deren Codierung werden in der Vergleichseinheit 14 miteinander verglichen. Im Falle des Nichtvorhandenseins eines Fehlers wird ein Freigabesignal 16 erzeugt. Dieses Freigabesignal 16 wird in die Freigabeeinrichtung 15 eingekoppelt, welche dazu veranlasst wird, die Ergebnisdaten auf einen Bus 4 zu schreiben. Über den Bus 4 können diese Ergebnisdaten dann wieder verarbeitet werden.

10

Das Freigabesignal 16 kann ferner dazu genutzt werden, die Register 8 bis 11 wieder freizuschalten, so dass die nächsten Operanden vom Bus 3 ausgelesen werden können und in den ALU-Einheiten 1, 2 verarbeitet werden können.

15

Mit der Anordnung in Figur 1 wird das Ergebnis nicht überprüft. Hier werden lediglich die Ergebnisdaten in der Vergleichseinheit 14 miteinander verglichen. Eine Überprüfung der ECC-Codierung der Ergebnisdaten wird erst durch die Anordnung in Figur 2 möglich, bei der sowohl die Ergebnisdaten als auch deren ECC-Codierung miteinander in der Vergleichseinheit 14 verglichen werden.

20

Mit den in den Figuren 1 und 2 angegebenen Fehlererkennungsanordnungen werden alle transienten Fehler, permanenten Fehler und auch Laufzeitfehler erkannt. Laufzeitfehler innerhalb einer ALU-Einheit 1, 2 werden erkannt, wenn das Ergebnis nicht oder zu spät zu der Vergleichseinheit 12 gelangt und somit ein Vergleich mit einem Teilergebnis erfolgt. Durch die Absicherung der Operandenregister 8, 9, 10, 11 mit Fehlererkennungs- und Fehlerkorrekturcode und dem Vergleich der Endergebnisse ist der jeweilige Fehlerort und Fehlerzeitpunkt genau zu lokalisieren. Somit kann auf eine transiente Störung sehr schnell reagiert werden.

25

30 Es ergeben sich somit die folgenden Möglichkeiten zur Fehlerlokalisierung:

Ergibt ein Vergleich der Ergebnisdaten in der Vergleichseinheit 14 einen Unterschied, dann kann auf einen Fehler innerhalb der ALU-Einheiten 1, 2 geschlossen werden.

Ergibt ein Vergleich der ECC-Codierung in einer der Vergleichseinheiten 5, 6 einen Unterschied, dann kann auf ein fehlerhaftes Signal von dem Bus 3 bzw. vorgeschalteten Komponenten geschlossen werden.

5      Ergibt ein Vergleich der ECC-Codierung in der Vergleichseinheit 14 einen Unterschied, dann kann auf eine fehlerhafte Codierung des Ergebnisses geschlossen werden.

10     Zur Umschaltung zwischen dem genannten Sicherheitsmodus, in dem eine redundante Abarbeitung und Prüfung erfolgt und dem Leistungsmodus, bei dem eine Leistungssteigerung durch separate Programmabarbeitung erreicht wird, dient eine Umschalteinrichtung UE 17. Durch diese Umschalteinrichtung 17 werden die Elemente 8, 9 und 1, 2 so geschaltet, dass im einen Fall, also im Sicherheitsmodus SM eine redundante Programmabarbeitung, insbesondere eine synchrone Programmabarbeitung erfolgt und im zweiten Betriebsmodus, dem Leistungsmodus LM eine parallele Abarbeitung unterschiedlicher Programme vollzogen werden kann. Dazu können Schalter oder Schaltmittel vorgesehen sein, die zum Einen in den Elementen 8, 9 respektive 1, 2 oder auch in der Umschalteinrichtung 17 lokalisiert sein können oder zusätzlich separat von den Elementen 8, 9, 1, 2 bzw. 17 in der Schaltung enthalten sind.

20     Zur Umschaltung erfolgt eine Kennzeichnung der Programme oder Aufgabenprogramme oder Programmteile, also Codeblöcke oder auch der Befehle durch eine Kennung, durch welche erkennbar ist, ob diese sicherheitsrelevant sind, also im Sicherheitsmodus SM abgearbeitet werden müssen oder dem Leistungsmodus LM zugänglich gemacht werden können. Dies kann durch ein Bit im Befehl erfolgen, oder es kann durch einen speziellen Befehl die darauffolgende Sequenz gekennzeichnet werden. Dies wird näher anhand der unterschiedlichen Kennzeichnungsmöglichkeiten bei Figur 5 noch einmal beschrieben.

25     Dabei können die Programme zum Einen Anwendungsfunktionen umfassen, also insbesondere z. B. zur Steuerung von Betriebsabläufen bei einem Fahrzeug vorhanden sein, oder aber die Umschaltung erfolgt bezüglich Programmen, bei denen die Kennzeichnung auf Betriebssystemebene erfolgt, also z. B. eine Zuordnung ganzer Betriebssystemtasks.

Bei einer Decodierung kann nun die Umschalteinrichtung 17 erkennen, ob die nun folgende Berechnung sicherheitsrelevant ist, also im Sicherheitsmodus ausgeführt werden soll oder nicht. Ist dies der Fall, so werden die Daten an beide Ausführungseinheiten 1 und 2 übergeben. Ist dies nicht der Fall, wird also im Leistungsmodus weitergearbeitet, so bekommt eine Ausführungseinheit die Daten bereitgestellt und gleichzeitig kann die nächste Anweisung, wenn diese ebenfalls nicht sicherheitsrelevant ist, an die zweite Ausführungseinheit gegeben werden, so dass die Programme parallel mit höherem Durchsatz abgearbeitet werden.

10 Im ersten Fall beispielsweise dauert die Berechnung des Ergebnisses bei synchroner Abarbeitung auf beiden Einheiten gleich lang. Die Ergebnisse stehen also im Sicherheitsmodus bei synchroner Abarbeitung gleichzeitig bereit. Diese Daten werden nun am Ausgang entsprechend 12 und 13 wieder mit einer Codierung versehen, und es werden die Daten und/oder die Codierung dieser Daten, wie in den Figuren 1 und 2 beschrieben, bei Result a und Result b verglichen. Stimmen sie überein, so werden die Daten freigegeben; ansonsten erfolgt eine der angesprochenen Fehlerreaktionen. Im zweiten Fall, also im Leistungsmodus LM, wenn die Daten parallel abgearbeitet werden, wird der Komparator oder Vergleicher 14 am Ausgang der beiden Rechenwerke nicht angesteuert und die Ergebnisse Result a und Result b werden nacheinander wieder in die Registerbank zurückgeschrieben und können auch nacheinander ausgegeben werden, wie es auch in einem superskalaren Prozessor der Fall ist.

25 Dieser erfindungsgemäße Umschaltvorgang wird noch einmal in den Figuren 3 und 4 erläutert. Dabei zeigt Figur 3 die Umschaltung aus dem Sicherheitsmodus in den Leistungsmodus und Figur 4 die Umschaltung vom Leistungsmodus in den Sicherheitsmodus.

30 Um von dem ersten Betriebsmodus, also hier dem Sicherheitsmodus oder Safetymode SM in den zweiten Betriebsmodus, also hier den Leistungsmodus oder Performancemode LM zu gelangen, ist eine Kennung und eine entsprechende Umschaltung erforderlich. In Figur 3 ist dies noch einmal verdeutlicht. In Block 300 ist die Ausführungseinheit 1 im zweiten Betriebsmodus, dem Leistungsmodus. Ebenso ist im Block 310 die zweite Ausführungseinheit 2 auch im Leistungsmodus. Gleichermassen werden durch die Umschalteinrichtung 17, die beispielsweise als Decoderbaustein ausgebildet ist,

respektive einen solchen enthält, die Elemente 8 und 9 gesteuert bzw. geschaltet. Im Block 320 bzw. Block 321 wird nun entsprechend des Programmablaufs der jeweiligen Ausführungseinheit 1 oder 2 wenigstens eine Kennung festgestellt, durch welche eine Umschaltung in Block 330 von beiden Ausführungseinheiten in den ersten 5 Betriebsmodus, den Sicherheitsmodus SM, erfolgt. Dadurch laufen beide Zweige über die Blöcke 8 und 9 und die Ausführungseinheiten 1 und 2 wieder redundant und insbesondere synchron bezüglich der durch die Kennung gekennzeichneten sicherheitsrelevanten Programme, so dass wieder der Sicherheitsmodus SM vorliegt. Dabei genügt es, dass in einem Programmablauf im Leistungsmodus, also in einem Zweig 10 eine solche Kennung zur Umschaltung vorhanden ist, um beide Ausführungseinheiten in den Sicherheitsmodus zu führen. Dabei muss unter Umständen noch die Abarbeitung des bereits angefangenen Programmcodes der anderen Ausführungseinheit abgearbeitet werden, um beide dann im Sicherheitsmodus weiterarbeiten zu lassen. Andererseits kann es auch vorgesehen sein, sofort in den Sicherheitsmodus zu wechseln und bei einem 15 nachfolgenden Leistungsmodus das angefangene Programm beginnend von der Unterbrechungsstelle weiter abzuarbeiten.

Um von dem ersten Betriebsmodus, also hier dem Sicherheitsmodus in den zweiten Betriebsmodus, den Leistungsmodus zu gelangen, wird nun entsprechend Figur 4 20 ebenfalls eine Kennung verwandt. Im Block 200 sind beide Ausführungseinheiten 1 und 2 und entsprechend die Zweige mit den Blöcken 8 und 9, also der Operandenzuschaltung im Sicherheitsmodus, dem ersten Betriebsmodus. In Abfrageblock 210 wird nun geprüft, ob eine Umschaltkennung vorliegt, respektive ob eine vorliegende Kennung die Umschaltung in den Leistungsmodus möglich macht. Ist dies nicht der Fall, liegt also 25 keine Kennung vor oder zeigt die Kennung weiterhin den Sicherheitsmodus an, gelangt man wieder zu Block 200 und die Programme werden weiter im Sicherheitsmodus abgearbeitet. Ist eine Kennung vorhanden respektive zeigt diese die Umschaltung an, erfolgt im Block 220 die Umschaltung bzw. der Wechsel in den zweiten Betriebsmodus, den Leistungs- oder Performancemode LM. Da im Sicherheitsmodus parallel, also 30 redundant, die gleichen Programme abgearbeitet werden, erfolgt hier eine Umschaltung nur, wenn für beide Zweige im Leistungsmodus, also Block 8 und ALU 1 sowie Block 9 und ALU 2 eine Umschaltung aufgrund der Kennung vorgesehen ist. Erfolgt eine vollsynchrone Abarbeitung, also eine zeitgleiche Abarbeitung des Programms, ist dies ohnehin gegeben, erfolgt eine nichtsynchrone Abarbeitung des Programms, muss von der

5 schnelleren Ausführungseinheit auf die zurückliegende gewartet werden, so dass die Umschalteinrichtung 17 erst dann umschaltet, wenn beide Kennungen vorliegen bzw. ausgewertet sind. Eine solche Synchronität muss auch zum Ergebnisvergleich bzw. ECC- und Ergebnisvergleich gemäß der Blöcke 12, 13 und 14 sowie 12a und 13a entweder durch Zeitgleichheit erzwungen oder durch Warten erzeugt werden.

10 In Block 230 ist somit dann der erste Zweig, also Block 8 und Ausführungseinheit 1 wieder im Leistungsmodus und im Block 231 der zweite Zweig mit Block 9 und Ausführungseinheit 2, wodurch dann die erfindungsgemäße Umschaltung vollzogen ist.

15 Damit ist entsprechend der Aufgabe eine optimierte Umschaltung zwischen zwei Betriebsmodi einer Prozessoreinheit mit zwei integrierten Ausführungseinheiten erfindungsgemäß dargestellt, wobei die Kennung auf verschiedenste Art und Weise entsprechend Figur 5 in einem Programm- oder Datenzeilenabschnitt 500 eingebracht oder lokalisiert sein kann. Im Weiteren sprechen wir bei den Zeilen in Figur 5 von Programmzeilen, wobei auch hier Programm- und Datenzeilen in beliebiger Kombination möglich sind.

20 So sind in Figur 5 beispielhaft die Programme P1 von Zeile Z1 bis Zeile Z6, P2 von Zeile Z7 bis Zeile Z15 und P3 von Zeile Z16 bis Zeile Z19 dargestellt. Mit AP ist ein Aufgabenprogramm beispielsweise als Teil eines Programmes P1 dargestellt, wobei auch mehrere Programme, z. B. P1 und P2, zusammen ein Aufgabenprogramm bilden können. Mit CB ist ein Codeblock dargestellt, also ein Programmteil, der beispielsweise Zeilen zweier Programme, hier Z14 bis Z18 der Programme P2 und P3 umfasst. Ebenso kann ein solcher Codeblock, also ein Programmteil nur Teil eines Programmes sein. Weiterhin ist mit PB3 entsprechend Zeile Z19 ein Programmbeahl dargestellt. Mit Zeilen ZS1 und ZS2 ist ein spezieller Speicherbereich SSB dargestellt, der als vorgegebener Speicherbereich eine solche Kennung, hier KB enthalten kann. Daneben sind mit K1, K2, K3 und K4 sowie KB verschiedene Kennungen dargestellt, die den unterschiedlichen Möglichkeiten des erfindungsgemäßen Verfahrens Rechnung tragen. Bezuglich des Einsatzes der Kennung gibt es nun verschiedene Möglichkeiten: Zum Einen kann als Grundverarbeitungsmodus, also als Default-Modus der Sicherheitsmodus SM vorgesehen sein (ebenso natürlich der Leistungsmodus). Bei Vorhandensein einer Kennung wird dann entsprechend in den Leistungsmodus umgeschaltet (oder umgekehrt in den

Sicherheitsmodus). Andererseits kann es erfindungsgemäß auch vorgesehen sein, dass grundsätzlich eine Kennung vorhanden sein muss und aus dem Inhalt der Kennung, also insbesondere deren Bitwertes auf den entsprechenden Modus geschlossen wird. Dabei ist dann beispielsweise ein Binärwert 1 (oder auch ein anderer Wert, insbesondere der dominante Wert) dem Sicherheitsmodus SM zugeordnet und der Binärwert 0 (oder auch ein anderer Wert insbesondere der rezessive Wert) dem Leistungsmodus LM. Bezuglich der Überlegung von dominant und rezessiv hat dies zur Folge, dass bei einem Fehler oder einem Ausfall in der Regel der dominante Wert und somit der Sicherheitsmodus eingestellt ist. Entsprechend Zeile Z4 ist nun mit Kennung K1 ein Binärwert B1, also K1/B1 vorhanden, welcher beispielsweise anzeigt, dass das Aufgabenprogramm der Zeilen Z4 bis Z6 im Programm P1 im Leistungsmodus abgearbeitet werden kann, obwohl beispielsweise Programm P1 im Sicherheitsmodus abgearbeitet werden muss. Wie den Kennungen K1, K2 und K3 anzusehen, können diese unterschiedlich lang sein, so dass beispielsweise bei Kennung K2 gemäß Zeile Z7 3 Bit, B1 bis B3, die Kennung ausmachen, so dass zum Einen mit Bit B1 bei K2 Sicherheitsmodus SM oder Leistungsmodus LM entschieden wird und beispielsweise die Bits B2 und B3 die Zeilenzahl angeben, für welche dieser Modus, beispielsweise der Sicherheitsmodus, gilt, so dass das gesamte Programm P2 oder auch nur ein Teil davon im Sicherheitsmodus abgearbeitet wird. Ebenso können Codeblöcke, also Programmteile, die beispielsweise keine Gesamtaufgabe umfassen, also kein Aufgabenprogramm darstellen, hier mit CB dargestellt, durch eine Kennung, wie hier K3 einem Modus zugeordnet werden. Hier kann dann neben der Betriebsmodzuordnung mit Bit B1 bei K3 beispielsweise auch eine Anfangszeile oder Adresse mit Bits B2 und B3 bei K3 und eine Endzeile oder Endadresse mit Bits B4 und B5 bei K3 angegeben sein, so dass ein spezieller Bereich in einem entsprechend zugeordneten Betriebsmodus abgearbeitet wird. Eine solche Kennungszuordnung kann gemäß K4 aber auch bei einzelnen Befehl PB3 in Z19 oder auch bei jedem Befehl erfolgen. Wie dargestellt, können also diese Kennungen vollständigen Programmen oder Aufgabenprogrammen AP oder Programmteilen CB, oder auch einzelnen Programmbefehlern PB, hier PB3, zugeordnet sein, was dann eine entsprechende Umschaltung durch die Umschalteinrichtung 17 auslöst. Durch die Abfrage in Block 210 oder auch in den Blöcken 320 und 321 wird dann das Vorhandensein einer solchen Kennung K1 bis K4 oder KB überprüft respektive deren Inhalt umgeschaltet. Dabei kann die Kennung, wie hier dargestellt, als wenigstens ein Bit ausgebildet sein, aber auch mehrere Bits umfassen, zum Einen abhängig von der

unterschiedlichen Anzahl der Betriebsmodi, zum Anderen bedingt durch Zusatzinformationen wie die Anzahl der Zeilen oder eine Anfangs- oder Endadresse.

5 In einer speziellen Ausführungsform kann wenigstens ein Programmbebefl vorgesehen sein, hier PB1, PB2 oder auch PB3, der eine Kennung erst erzeugt, die anzeigt, ob im ersten oder zweiten Betriebsmodus abzuarbeiten ist. Dabei kann die Kennung in einem bestimmten Speicherbereich SSB eingeschrieben werden, wie hier mit KB in ZS2 dargestellt. Dieser Bereich SSB kann sich in einem Register, in einem in der CPU integrierten Speicher, aber auch in einem dazu externen Speicher befinden. Dabei kann 10 als diese Kennung KB erzeugender Befehl ein Spezialbefehl, z. B. PB3 vorgesehen sein oder aber ein Befehl, der bereits im Befehlssatz der Prozessoreinheit vorhanden ist. Als Spezialbefehl kann somit beispielsweise eben ein Befehl "erzeuge Kennung" implementiert werden, oder es kann auf einen im Prozessorbefehlssatz bereits vorhandenen Befehl, insbesondere einen Schreibbefehl zurückgegriffen werden, wie hier 15 durch PB1 und PB2 dargestellt, so dass in Z9 der Schreibbefehl WR in den Speicherbereich KB den Binärwert 0 einschreibt, dargestellt durch WR (KB: 0) und damit alle nachfolgenden Zeilen, solange die Kennung KB0 ist, z. B. im Sicherheitsmodus abgearbeitet werden. Mit dem selben Befehl kann dann in Z12 bei PB2 durch WR (KB: 1) in den Speicherbereich für die Kennung KB der Wert 1 eingegeben 20 werden, so dass ab diesem Zeitpunkt die nachfolgenden Zeilen, z. B. im Leistungsmodus, abarbeitbar sind. D. h. durch einfache kennungserzeugende Befehle insbesondere ein simpler Schreibbefehl WR kann dann beispielsweise in einem speziellen Speicherbereich, der regelmäßig abgefragt wird, eine entsprechende Umschaltkennung KB erzeugt werden.

25

Damit sind eine Vielzahl erfundungsgemäßer Möglichkeiten dargestellt, eine Betriebsmodumschaltung bei einer Prozessoreinheit mit zwei Ausführungseinheiten aufgrund einer Kennung durchzuführen. Damit sind die genannten Vorteile der Erfindung erzielbar.

29.09.03 Sy/Ho

5

ROBERT BOSCH GMBH, 70442 Stuttgart

10

Ansprüche

15

1. Verfahren zur Umschaltung zwischen wenigstens zwei Betriebsmodi (SM, LM) einer Prozessoreinheit (100, 101) mit wenigstens zwei Ausführungseinheiten (ALUA, ALUB) zur Abarbeitung von Programmen (P1, P2, P3), dadurch gekennzeichnet, dass wenigstens den Programmen (P1, P2, P3) wenigstens eine Kennung (K2) zugeordnet ist, welche eine Unterscheidung in die wenigstens zwei Betriebsmodi (SM, LM) erlaubt und eine Umschaltung zwischen den Betriebsmodi in Abhängigkeit von der Kennung (K1-K4, KB) erfolgt, so dass die Prozessoreinheit (100, 101) die Programme (P1, P2, P3) entsprechend des zugeordneten Betriebsmodus abarbeitet.

20

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Programme (P1, P2, P3) Aufgabenprogramme (AP) enthalten oder diese bilden und den einzelnen Aufgabenprogrammen (AP) jeweils die Kennung (K1) zugeordnet ist.

25

3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Programme (P1, P2, P3) aus einzelnen Programmteilen (CB) bestehen oder diese enthalten und den einzelnen Programmteilen (CB) jeweils die Kennung (K3) zugeordnet ist.

30

4. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Programme (P1, P2, P3) aus einzelnen Programmbefehl (PB) bestehen und den einzelnen Programmbefehl (PB) jeweils die Kennung (K4) zugeordnet ist.

5. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Programme (P1, P2, P3) zu einem Betriebssystem der Prozessoreinheit (100, 101) gehören oder das

Betriebssystem darstellen.

6. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Programme (P1, P2, P3) zur Steuerung von Betriebsabläufen eines Fahrzeugs eingesetzt werden.

5

7. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass ein erster Betriebsmodus vorgesehen ist, der einem Sicherheitsmodus (SM) entspricht, bei dem die zwei Ausführungseinheiten (ALUA, ALUB) gleiche Programme (AP, P2) redundant abarbeiten.

10

8. Verfahren nach Anspruch 7, dadurch gekennzeichnet, dass bei der Abarbeitung der Programme (AP, P2) entstehende Zustände oder Ergebnisse (ResultA, ResultB) auf Übereinstimmung verglichen (14) werden und bei Abweichung auf Fehler erkannt wird.

15

9. Verfahren nach Anspruch 7, dadurch gekennzeichnet, dass die Programme (AP, P2) synchron abgearbeitet werden.

20

10. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass im zweiten Betriebsmodus, der einem Leistungsmodus (LM) entspricht, jede Ausführungseinheit (ALUA, ALUB) unterschiedliche Programme (P1, P2, P3) abarbeitet.

25

11. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Kennung (K1, KB) als wenigstens ein Bit (K1/B1) ausgebildet ist.

30

12. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass ein Programmbeahl (PB1, PB2, PB3) vorgesehen ist der eine Kennung (KB) erzeugt, die anzeigt, ob im ersten oder zweiten Betriebsmodus abzuarbeiten ist.

13. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Kennung (KB) in einem bestimmten Speicherbereich (SSB) eingeschrieben ist.

14. Verfahren nach Anspruch 12 oder 13, dadurch gekennzeichnet, dass die Kennung (KB) durch einen in einem Befehlssatz der Prozessoreinheit vorgesehenen Befehl (PB1, PB2) erzeugt wird.

15. Verfahren nach einem der Ansprüche 12-14, dadurch gekennzeichnet, dass die Kennung (KB) durch einen Schreibbefehl (WR) erzeugt wird.

5 16. Vorrichtung zur Umschaltung zwischen wenigstens zwei Betriebsmodi (LM, SM) einer Prozessoreinheit (100, 101) zur Abarbeitung von Programmen (P1, P2, P3) mit wenigstens zwei Ausführungseinheiten (ALUA, ALUB), wobei Umschaltmittel (8, 9) enthalten sind, durch welche umgeschalten werden kann, dadurch gekennzeichnet, dass diese Umschaltmittel (8, 9) wenigstens den Programmen (P1, P2, P3) wenigstens eine Kennung (K1-K4, KB) zuordnen, welche eine Unterscheidung in die zwei Betriebsmodi (LM, SM) erlaubt und die Umschaltmittel (8, 9) derart ausgestaltet sind, dass diese in Abhängigkeit von der Kennung zwischen den Betriebsmodi umschalten und die Prozessoreinheit die Programme entsprechend des zugeordneten Betriebsmodus abarbeitet.

10 17. Vorrichtung nach Anspruch 16, dadurch gekennzeichnet, dass als wenigstens zwei Ausführungseinheiten entsprechend wenigstens duplizierte Rechenwerke (ALUA, ALUB) vorgesehen sind.

15 18. Prozessoreinheit (100, 101) zur Abarbeitung von Programmen (P1, P2, P3) mit wenigstens zwei Ausführungseinheiten (ALUA, ALUB), wobei Umschaltmittel (8, 9) enthalten sind, durch welche zwischen wenigstens zwei Betriebsmodi (LM, SM) der Prozessoreinheit umgeschalten werden kann, dadurch gekennzeichnet, dass diese Umschaltmittel (8, 9) wenigstens den Programmen (P1, P2, P3) wenigstens eine Kennung (K1-K4, KB) zuordnen, welche eine Unterscheidung in die zwei Betriebsmodi (LM, SM) erlaubt und die Umschaltmittel (8, 9) derart ausgestaltet sind, dass diese in Abhängigkeit von der Kennung zwischen den Betriebsmodi umschalten und die Prozessoreinheit die Programme entsprechend des zugeordneten Betriebsmodus abarbeitet.

29.09.03 Sy/Ho

5

ROBERT BOSCH GMBH, 70442 Stuttgart

10

Verfahren und Vorrichtung zur Umschaltung zwischen wenigstens zwei Betriebsmodi einer Prozessoreinheit

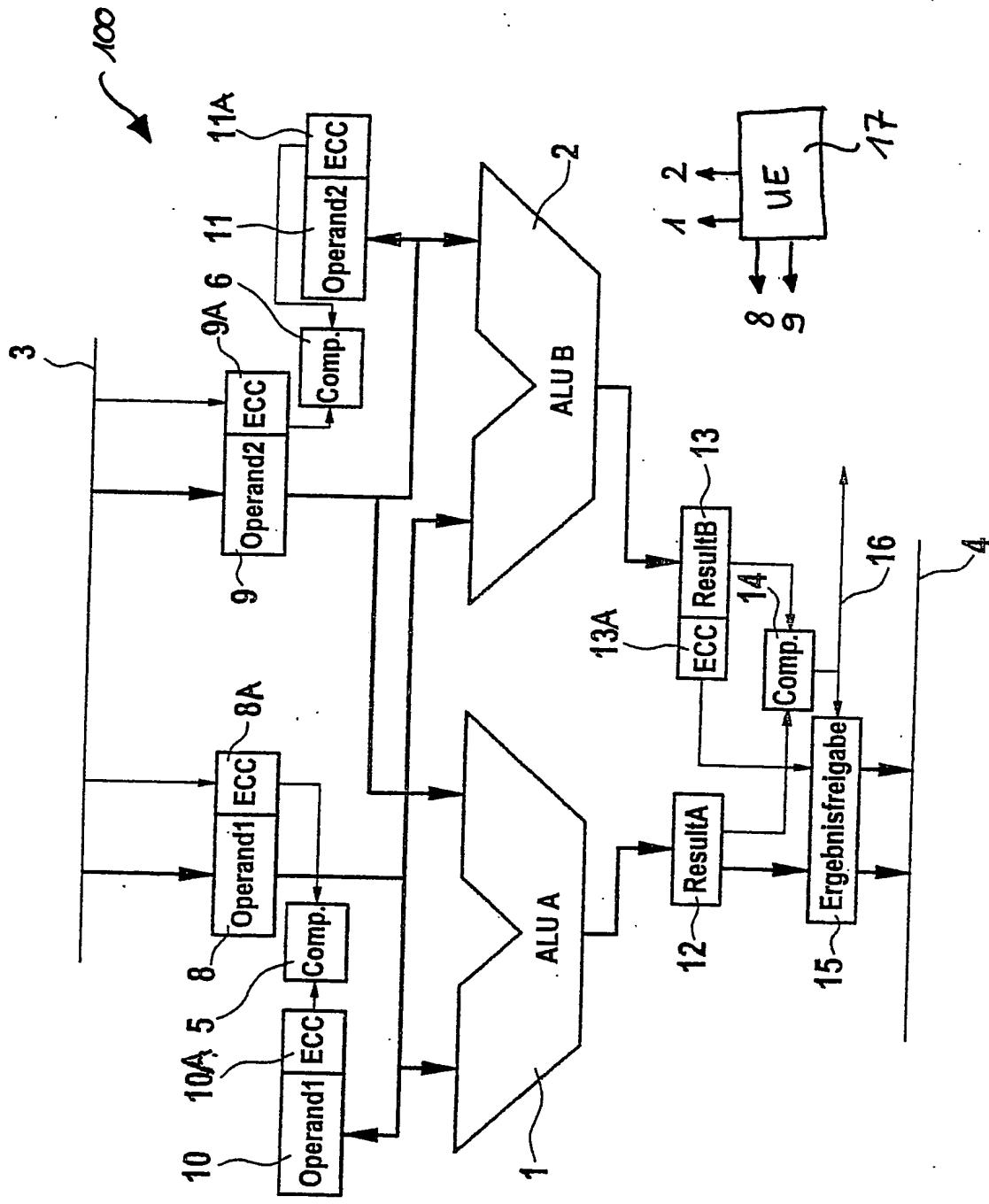
Zusammenfassung

15

Verfahren und Vorrichtung zur Umschaltung zwischen wenigstens zwei Betriebsmodi (SM, LM) einer Prozessoreinheit (100, 101) mit wenigstens zwei Ausführungseinheiten (ALUA, ALUB) zur Abarbeitung von Programmen (P1, P2, P3), wobei wenigstens den Programmen (P1, P2, P3) wenigstens eine Kennung (K2) zugeordnet ist, welche eine Unterscheidung in die wenigstens zwei Betriebsmodi (SM, LM) erlaubt und eine Umschaltung zwischen den Betriebsmodi in Abhängigkeit von der Kennung (K1-K4, KB) erfolgt, so dass die Prozessoreinheit (100, 101) die Programme (P1, P2, P3) entsprechend des zugeordneten Betriebsmodus abarbeitet.

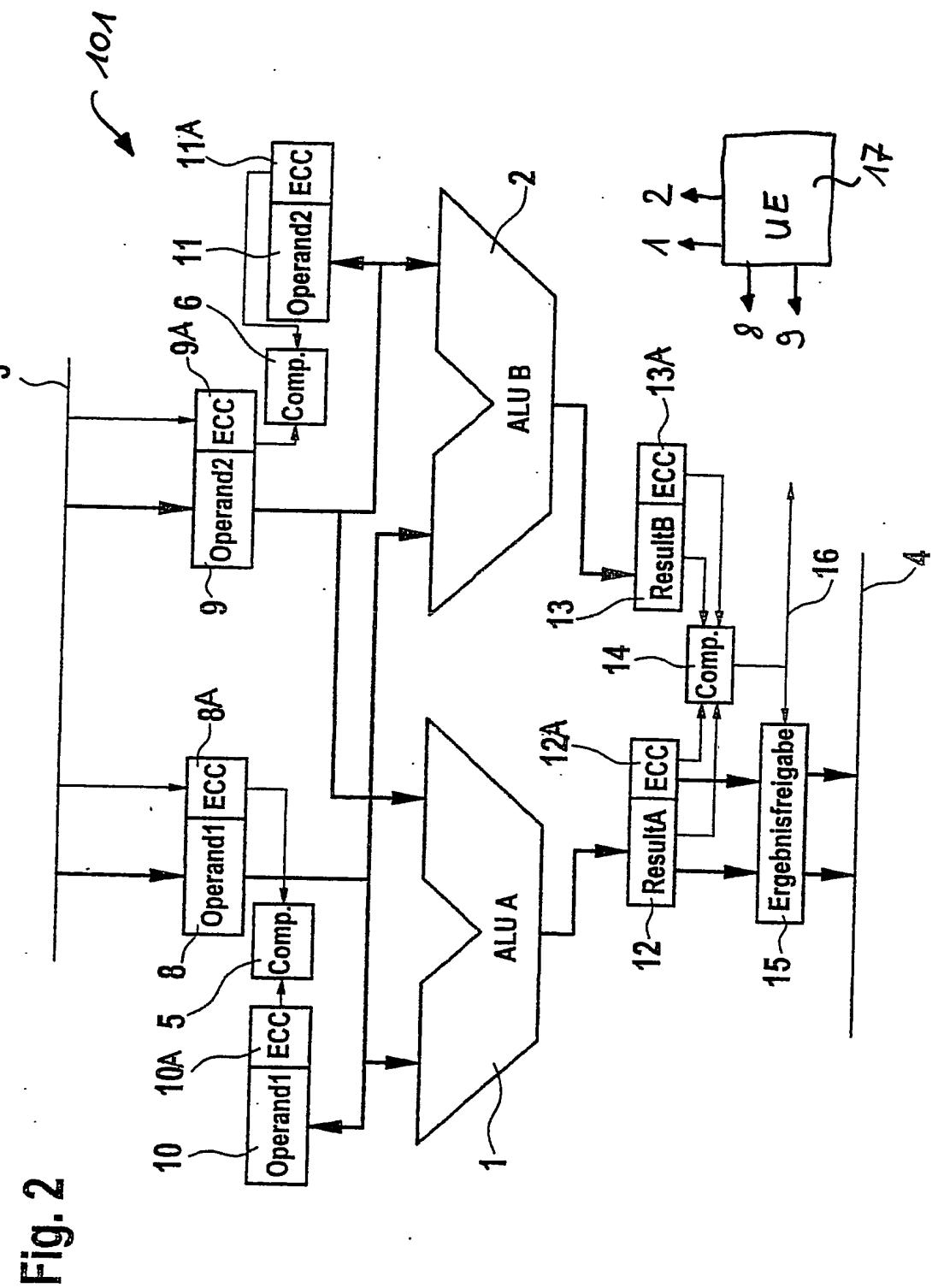
20

Fig. 1



1 / 5

R. 306858



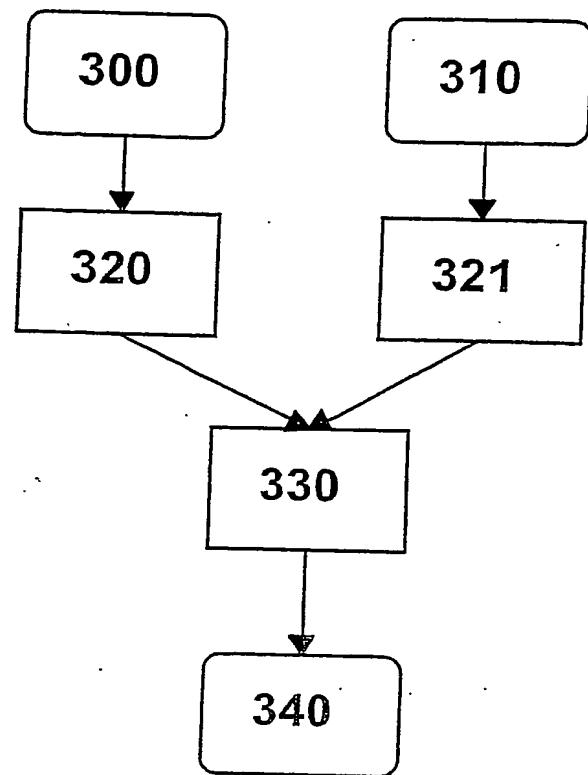


Fig. 3

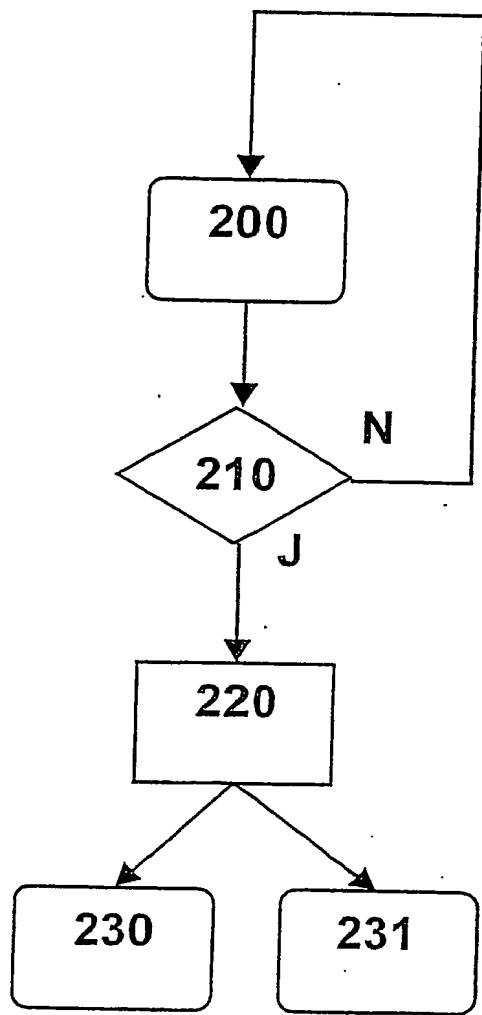


Fig. 4

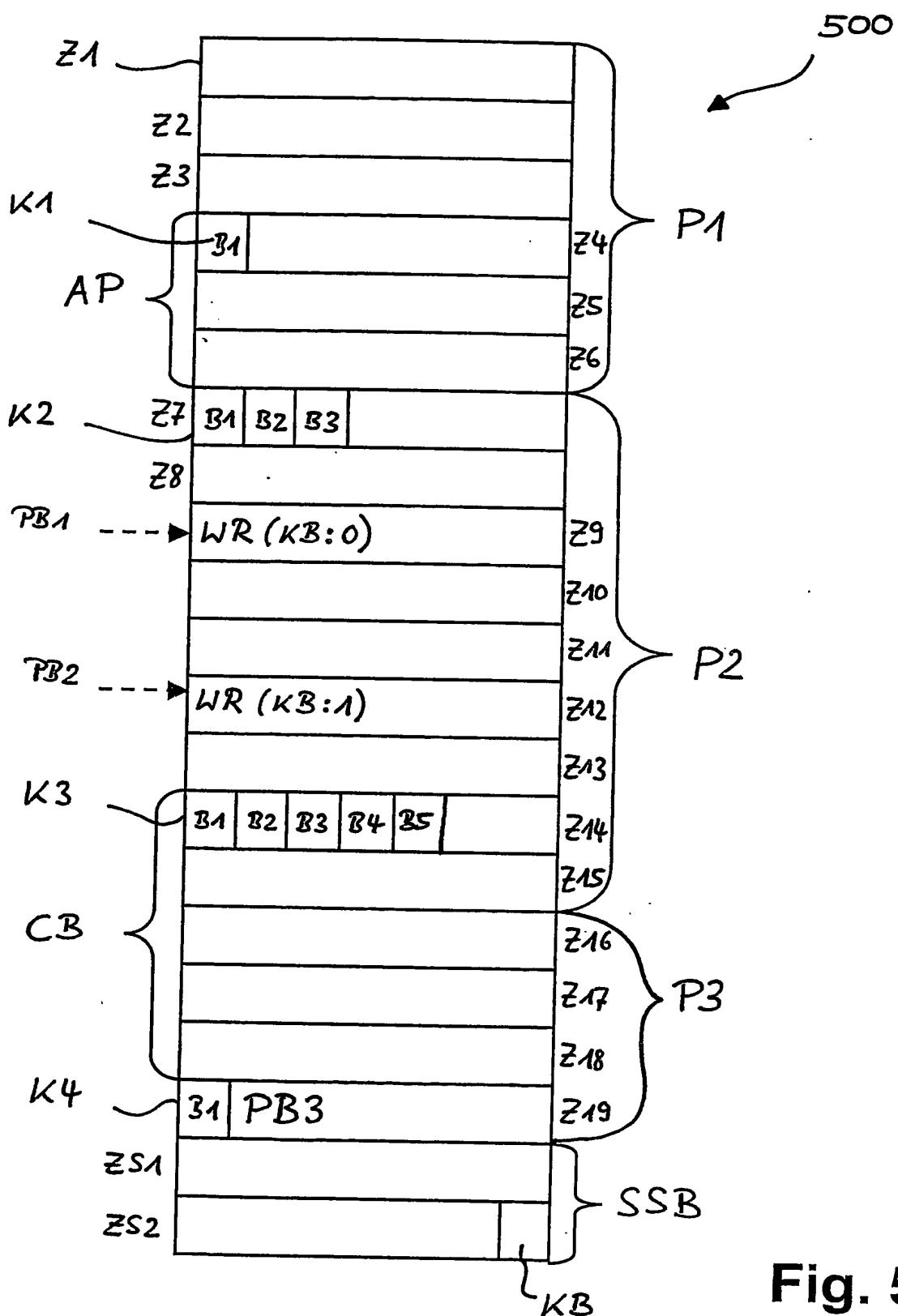


Fig. 5